

Tesina de Grado
para la obtención del título de
Licenciado en Ciencias de la Computación

Seguridad en iOS y Android: un Análisis Comparativo

Autor

Raúl Ignacio Galuppo

raul.i.galuppo@gmail.com

G-3483/5

Director

Dr. Carlos Luna



Departamento de Ciencias de la Computación
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario
Diciembre de 2017

Índice general

1. Hacia un Framework Comparativo	2
1.1. Vista principal	2
1.1.1. Funciones no compatibles	4

Capítulo 1

Hacia un Framework Comparativo

Android e iOS permiten cambiar ciertos permisos de una aplicación en tiempo de ejecución, es decir, luego de haberla instalado en el dispositivo. Para poner a prueba los sistemas de permisos de ambas plataformas, se ha desarrollado un framework que permite comparar formalmente diversos aspectos de seguridad.

El framework es una aplicación móvil y esta compuesto por varios tests. Cada test pone a prueba a un componente del dispositivo, permitiendo así conocer el alcance de los permisos correspondientes a dicho componente.

Todos los test fueron implementados en *JavaScript* y corren en el dispositivo mediante *Apache Cordova*¹.

En las siguientes secciones se detallaran los distintos tests que componen el framework. Además, se mencionarán las conclusiones arribadas luego de correr los tests mencionados anteriormente.

1.1. Vista principal

Al iniciar la aplicación, lo primero que se observa son observan dos áreas principales: Acciones y Test.

La primera área contiene un botón para acceder a la configuración de los permisos del dispositivo. Allí, el *tester* puede cambiar manualmente los permisos requeridos por la aplicación. Además, se encuentra un botón para limpiar la consola (que se encuentra en el otro área).

Mientras que la segunda área se subdivide en dos: en la parte de los tests y la parte de la consola. Una parte corresponde a los botones de los tests que, al presionarse, ejecutan el respectivo test, mostrando en la consola el resultado. Dicho resultado se mostrara con tipografía color verde si fue

¹TODO: va la referencia hacia donde se explica que es Apache Cordova.

exitoso; en cambio, se mostrara con tipografía color roja de ser fallido.
En la Figura 1.1 se observan las áreas del framework.

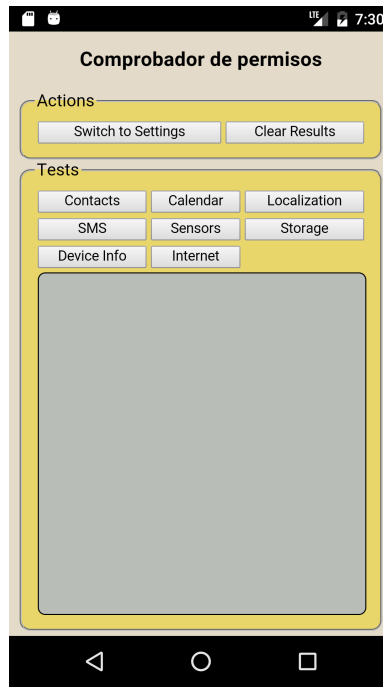


Figura 1.1: Áreas de la aplicación *Runtime Permissions Test*.

A continuación se mencionan los componentes que se pueden testear con el framework:

- Contactos
- Calendario
- Localización
- SMS
- Sensores
- Almacenamiento
- Información del dispositivo.
- Acceso a Internet

1.1.1. Funciones no compatibles

El emulador oficial de Android es compatible con la mayoría de las funciones de un dispositivo, pero no incluye la posibilidad de virtualizar los siguientes componentes [1]:

- WiFi;
- Bluetooth;
- NFC²;
- Manipulación de la tarjeta SD;
- Conexión USB;
- Micrófono;
- Cámara

Al no poder manipular la tarjeta SD, no es posible testear ninguna de las funcionalidades multimedia: no se pueden grabar audio, ni video ni sacar fotos.

Por lo tanto, no se agregaron al framework tests para los componentes listados anteriormente.

1.2. Catálogo de test

En esta sección se listaran todos los test que conforman al framework. Para cada test se detallara su algoritmo, los plugins de Apache Cordova que se utilizaron para confeccionarlo y una serie de capturas.

1.2.1. Calendario

Plugin: cordova-plugin-calendar v4.5.5

Algorithm 1 Test de los permisos del calendario

- 1: se crea la fecha *startDate*
 - 2: se crea la fecha *endDate*
 - 3: se crea un evento que empieza en la fecha *startDate* y termina en la fecha *endDate*.
 - 4: se listan los eventos entre las fechas *startDate* y *endDate*.
-

²Del ingles *Near Field Communication*. Es una tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos.

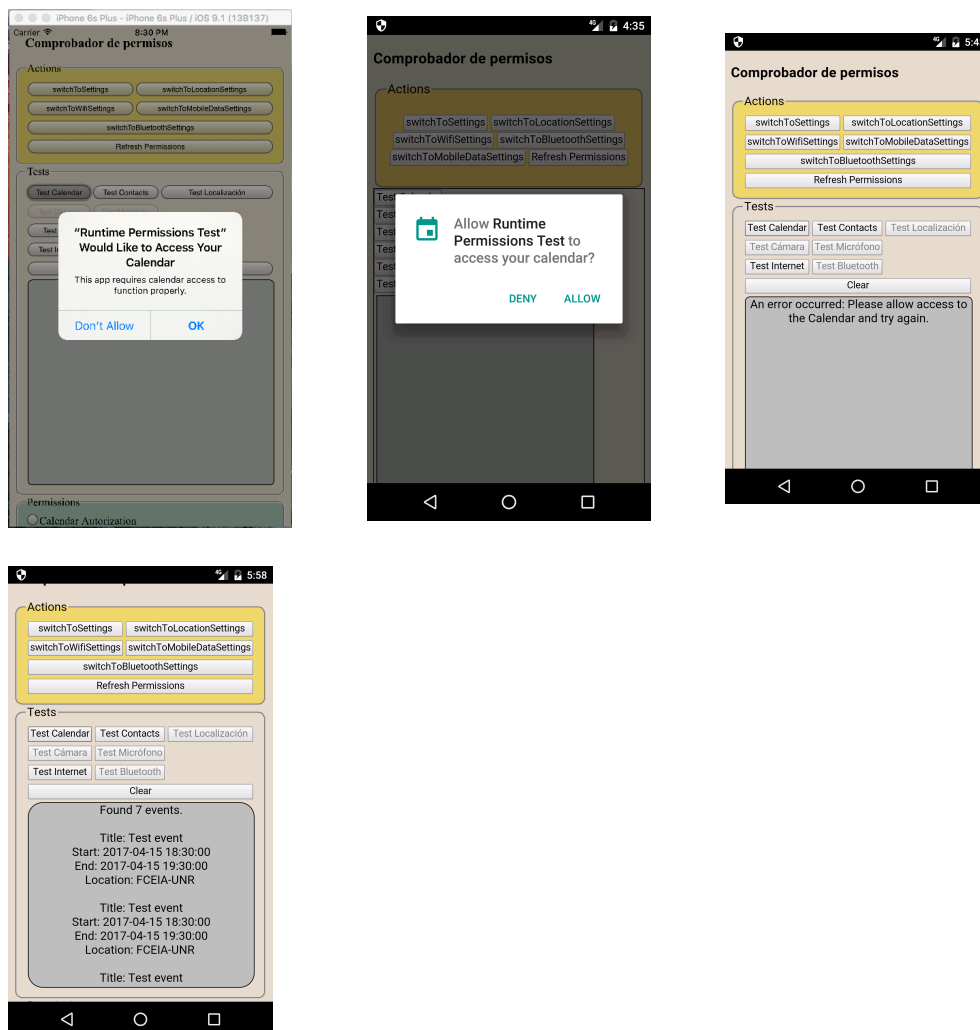


Figura 1.2: Testeando el calendario

Contactos

Plugin: cordova-plugin-contacts v2.2.1

Algorithm 2 Test de los permisos de los contactos

- 1: se listan todos los contactos.
 - 2: se crea un nuevo contacto.
 - 3: se listan todos los contactos.
-

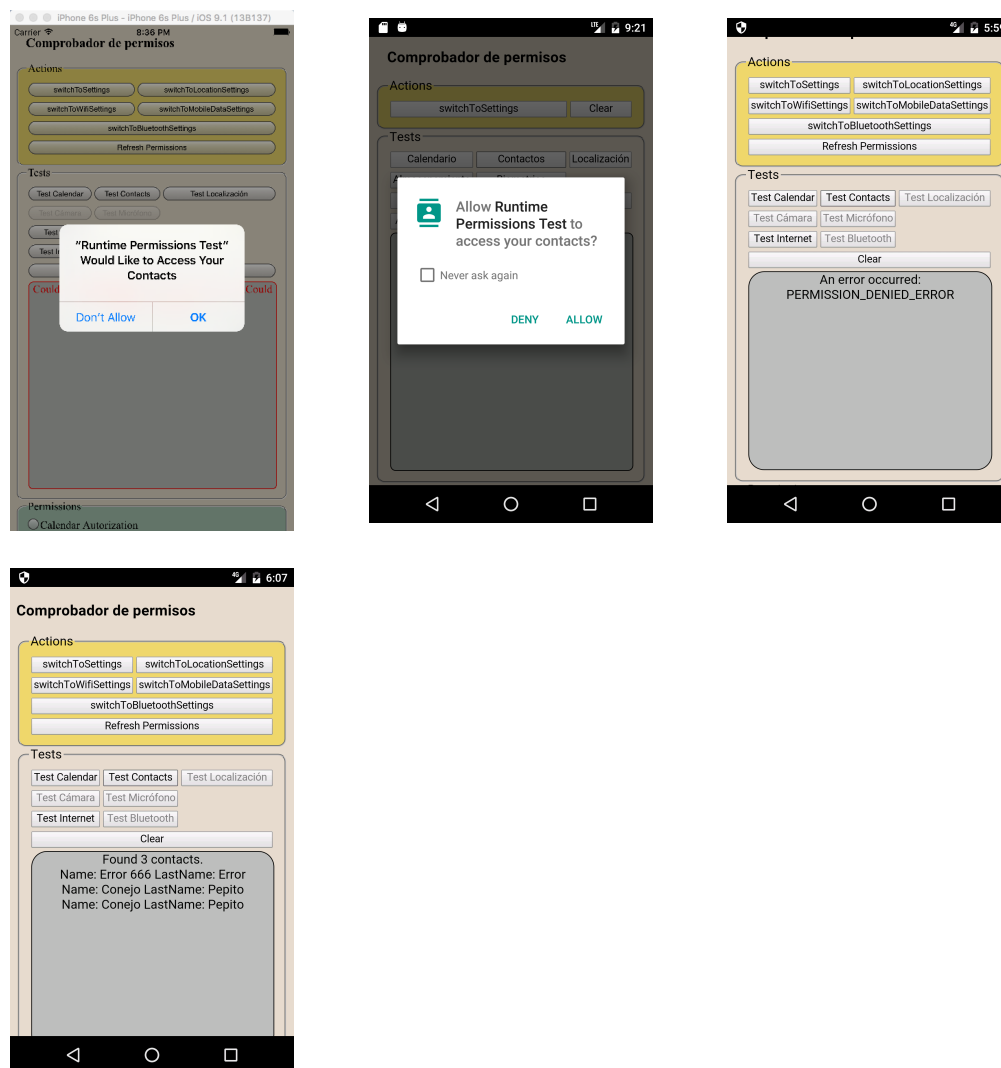


Figura 1.3: Testeando la administración de los contactos

Geolocalización

Plugin: cordova-plugin-geolocation v2.4.3

Desde el emulador, se simula unas coordenadas (Figura 1.4). Mediante el plugin utilizado, se obtienen los datos, siempre y cuando se tengan los

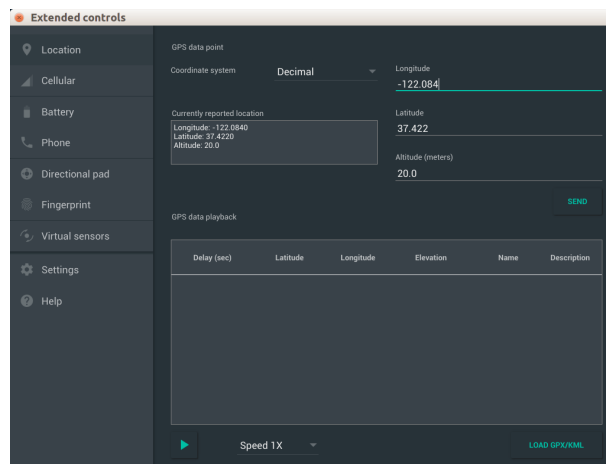


Figura 1.4: Se pueden observar las coordenadas que se emulan.

permisos correspondientes. En la Figura 1.5 se observa lo mencionado anteriormente. El test simplemente consiste en pedir las coordenadas actuales.

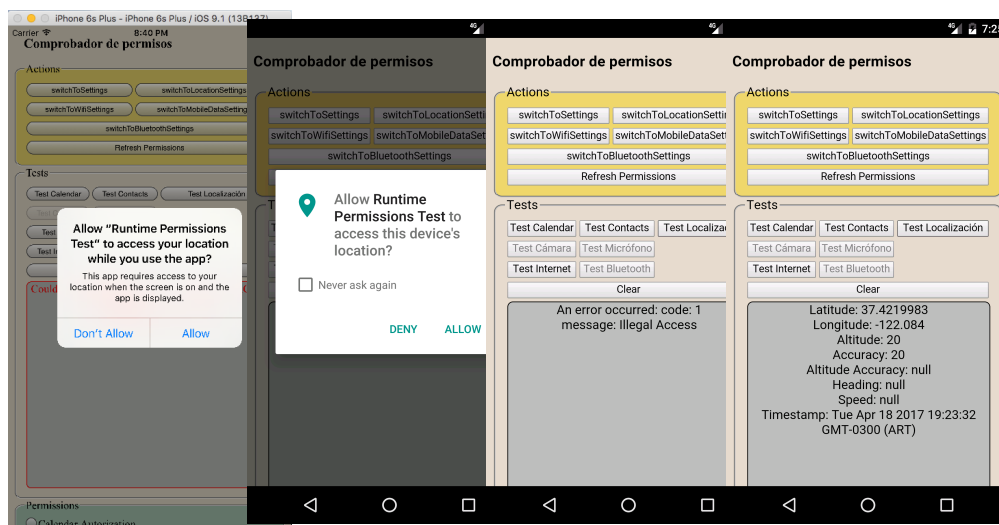


Figura 1.5: Testeando la geolocalización

Almacenamiento

En iOS no fueron necesarios permisos para poder correr el test. *Plugin:* cordova-screenshot

Algorithm 3 Test de los sensores

- 1: Se intenta capturar la pantalla y guardarla en el dispositivo.
-

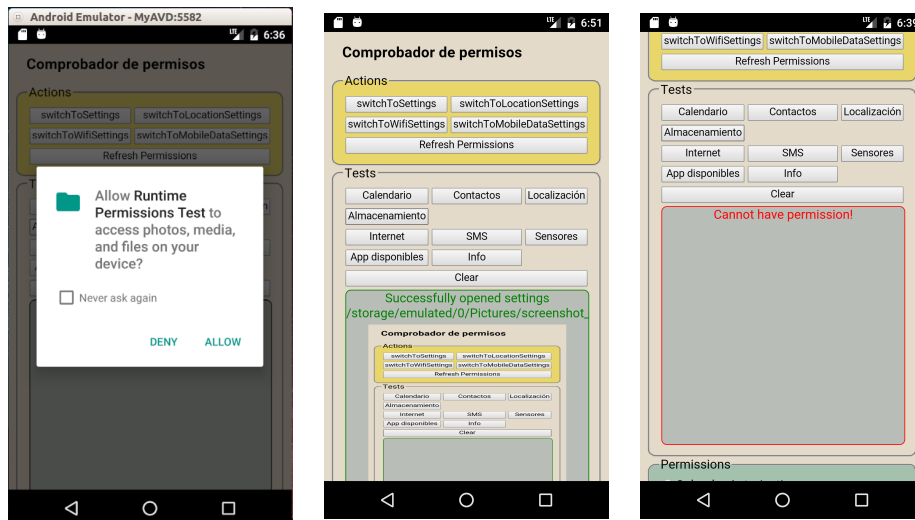


Figura 1.6: Testeando el almacenamiento.

SMS

Plugin: cordova-plugin-sms v.1

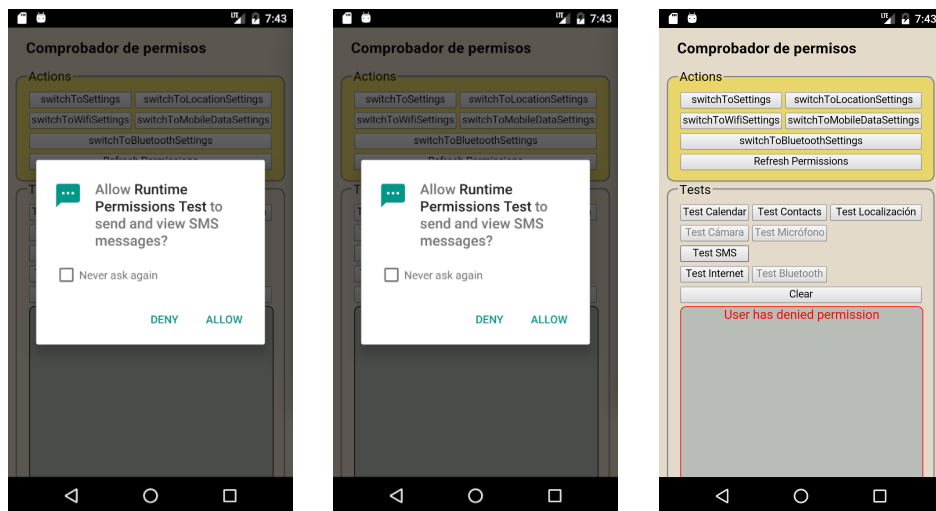
En un principio, se quería agregar al test la funcionalidad de leer los mensa-

Algorithm 4 Test de los permisos de los mensajes SMS

- 1: Se inicializan los eventos para recibir SMS.
 - 2: Se envia un SMS de prueba.
-

jes. Se decidió quitarla ya que a partir de iOS 8 no se pueden acceder a dichos mensajes desde una app instalada por el usuario citeforumdeveloperapple. En cambio, en Android si se pueden acceder, siempre que se tengan los permisos correspondientes.

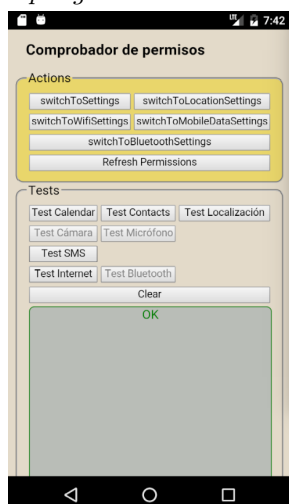
Internet



(a) 1^{era} vez pregunta si puede acceder a un permiso *peligroso*

(b) 1^{era} vez pregunta si puede acceder a un permiso *peligroso*

(c) Mensaje cuando no se tiene el permiso correspondiente.



(d) Mensaje luego de enviar un sms.

Figura 1.7: Testeando los mensajes SMS.

La decodificación de la imagen se obtuvo de StackOverflow. Al estar en un emulador, para probar el acceso a internet, se habilitó/deshabilitó la Red Inalámbrica.

Algorithm 5 Test de conexión a internet

- 1: se realiza una consulta GET HTTP hacia logo del DCC
 - 2: se decodifica la imagen (viene codificada en Base64).
 - 3: **return** un tag cuyo *source* es el dato decodificado.
-



(a) Respuesta satisfactoria a la consulta.
(b) Respuesta errónea a la consulta HTTP.

Figura 1.8: Testeando el acceso a internet

Sensors

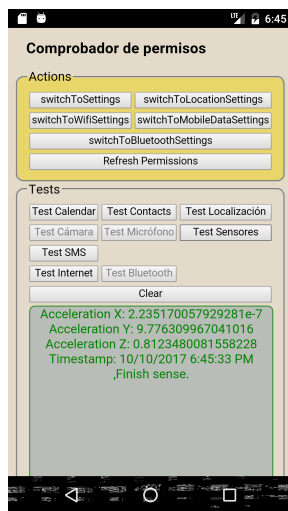
Plugin: cordova-plugin-device-motion

Plugin: cordova-plugin-gyroscope

En iOS no fueron necesarios permisos para poder correr el test.

Algorithm 6 Test de los sensores

- 1: Se inicializa un **TIMER** con **5 seg** para detener las mediciones.
 - 2: Se inicia la medición del acelerómetro.
 - 3: Se inicia la medición del giroscopio.
-



(a) Datos medidos

Figura 1.9: Testeando los sensores.

AppAvailable

Plugin: cordova-plugin-appavailability

TODO las capturas estan hechas pero no hay nada escrito No fueron necesarios permisos para poder correr el test.

DeviceInfo

Plugin: cordova-plugin-device

TODO las capturas estan hechas pero no hay nada escrito No fueron necesarios permisos para poder correr el test.

Batery

Plugin: cordova-plugin-battery-status

TODO las capturas estan hechas pero no hay nada escrito No fueron necesarios permisos para poder correr el test.

Health

Plugin: com.telerik.plugins.healthki

TODO las capturas estan hechas pero no hay nada escrito No fueron necesarios permisos para poder correr el test.

Bibliografía

- [1] ANDROID, D. *Developer Android: Funciones no compatibles*. <https://developer.android.com/studio/run/emulator.html#about>.